

[0038] As described above, according to the present embodiment, information, which indicates whether a task switching is performed by converting from an assembler language or a high-level language, is saved in an area 402 of a context preservation part 31. In
5 a case where the task switching is performed by converting from the high-level language, a saving of destructive registers is omitted, thereby optimizing an amount of context information to be saved in the context preservation part 31. Thus, it becomes possible to reduce a processing time for saving and restoring a
10 context and a usage of the context preservation part 31. Furthermore, an unused area generated by performing word alignment for a register having a length of three bytes, is assigned in the area 402. Thus, the usage increase, caused by a saving of identification data, of the context preservation part 31 is
15 prevented.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-134202

(43) 公開日 平成11年(1999) 5月21日

(51) Int.Cl. ⁶	識別記号	F I
G 0 6 F 9/46	3 4 0	G 0 6 F 9/46
	3 1 3	3 4 0 B
9/45		3 1 3 A
		9/44
		3 2 2 A

審査請求 未請求 請求項の数 9 O L (全 9 頁)

(21) 出願番号 特願平9-295134

(22) 出願日 平成9年(1997) 10月28日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 森本 浩

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 野村 琢家

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

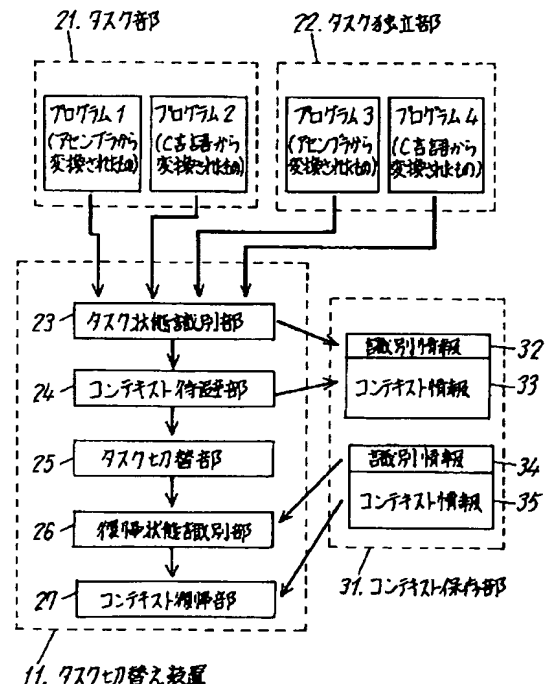
(74) 代理人 弁理士 滝本 智之 (外1名)

(54) 【発明の名称】 タスク切替え装置

(57) 【要約】

【課題】 従来のタスク切替え装置11は、タスク呼出し時のタスク状態の区別がつかないため、全タスク状態の中で必要な最大量のコンテキスト情報33を保存している。タスク状態では不要なコンテキスト情報33を保存するため、コンテキスト保存部31の使用量は増加する。本発明はタスク切替え時にタスク状態を保存し、コンテキスト保存部31の使用量を最小限にすることを目的とする。

【解決手段】 タスク切替え呼び出し時のタスク状態を保持するため、タスク状態識別部23がコンテキスト保存部31の特定領域にタスク状態の識別情報32を保存し、タスク切替え処理後、復帰状態識別部26がコンテキスト保存部31の識別情報32からタスク切替え前のタスク状態を識別できることで、コンテキスト保存部31のコンテキスト情報33を最適にする事ができ、コンテキスト保存部31の使用量を削減することができる。



【特許請求の範囲】

【請求項 1】 退避が必要なコンテキストを識別し、識別情報を所定の領域に退避するタスク状態識別手段と、前記タスク状態識別手段により識別されたコンテキストを退避するコンテキスト退避手段と、前記所定の領域から識別情報を読み出して復帰が必要なコンテキストを識別する復帰状態識別手段と、前記復帰状態識別手段により識別されたコンテキストを復帰するコンテキスト復帰手段とを備えることを特徴とするタスク切替え装置。

【請求項 2】 前記タスク状態識別手段は、タスク切り替えがアセンブラ言語のプログラムでなされたか高級言語のプログラムでなされたかの情報を前記識別情報とすることを特徴とする請求項 1 記載のタスク切替え装置。

【請求項 3】 前記コンテキスト退避手段は、前記タスク状態識別手段によりタスク切り替えが高級言語のプログラムでなされたことが識別された場合、前記高級言語の言語処理系において破壊レジスタと定義されたレジスタを含まないコンテキストを退避することを特徴とする請求項 2 記載のタスク切替え装置。

【請求項 4】 前記コンテキスト退避手段は、引数または戻り値を格納するレジスタに対して、該レジスタが前記破壊レジスタと定義されたレジスタに含まれる場合であっても、該レジスタを退避することを特徴とする請求項 3 記載のタスク切替え装置。

【請求項 5】 前記コンテキスト退避手段は、前記タスク状態識別手段によりタスク切り替えが高級言語のプログラムでなされたことが識別された場合であっても、該タスク切替えが各タスクから独立なプログラムからなされた場合は、前記破壊レジスタと定義されたレジスタを退避することを特徴とする請求項 3 記載のタスク切替え装置。

【請求項 6】 各タスクから独立な前記プログラムは、割り込みプログラムであることを特徴とする請求項 5 記載のタスク切替え装置。

【請求項 7】 前記識別情報は、退避または復帰するコンテキストにいかなる資源を含むかを、ビット毎に該資源を割り付けた情報または含む資源の状態をエンコードした情報であることを特徴とする請求項 1 記載のタスク切替え装置。

【請求項 8】 前記所定の領域と、前記コンテキスト退避手段によりコンテキストが退避される領域とは、予め定められたコンテキスト保存領域において互いに連続することを特徴とする請求項 1 から 7 のいずれか 1 項に記載のタスク切替え装置。

【請求項 9】 前記所定の領域は、前記コンテキスト退避手段によりコンテキストが退避される際に、該コンテキストに含まれる資源を所定のデータ長に整列させて退避するときに発生する未使用領域の一部または全部であることを特徴とする請求項 1 から 7 のいずれか記載のタ

スク切替え装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マイクロプロセッサ等に組み込まれるタスク切替え装置に関するものである。

【0002】

【従来の技術】図 7 は、従来のタスク切替え装置のブロック図である。

10 【0003】同図において、タスク切替え装置 111 は、コンテキスト待避部 123 とタスク切替部 124 とコンテキスト復帰部 125 から構成され、コンテキスト保存部 131 との間でデータの読み書きを行う。また、タスク部 121 には通常のプログラムを、タスク独立部 122 には割り込みプログラムを格納する。そしてプログラムは機械語であるが、アセンブラから変換されたもの（プログラム 1、3）と、C 言語等の高級言語から変換されたもの（プログラム 2、4）とがある。

20 【0004】タスク切替え装置 111 は、タスク部 121 またはタスク独立部 122 が現在実行しているタスク（プログラム）と別のタスクを実行しようとするとき、タスク切替え処理を行った後に、任意のタスクが実行される。従って、タスク部 121 のタスクが実行中にタスク独立部 122 のタスクが割り込んで実行される場合（割り込み）と、タスク独立部 122 のタスクが実行中にさらにタスク独立部 122 のタスクが割り込んで実行される場合（多重割り込み）とがある。また、割り込みが実行終了後はもとのタスクの実行が再開されることになる。

30 【0005】以上の構成を持つ従来のタスク切替え装置 111 について、コンテキスト保存部 131 にタスク部 121 とタスク独立部 122 とを実行させる為に必要なコンテキスト情報 141 を保存・復元する動作を説明する。

（1）コンテキスト保存部 131 へのコンテキスト情報 141 の待避動作

40 タスク部 121 又はタスク独立部 122 が現在実行中のタスクと別のタスクを実行しようとするとき、タスク切替え装置 111 はタスク状態に関係なくすべての一定量のコンテキスト情報 141 をコンテキスト保存部 131 に保存する。なお、コンテキスト情報 141 とはもとのタスクの実行するためのすべての情報をいい、PSW（プログラムステータスワード）等が含まれる。

（2）コンテキスト保存部 131 からのコンテキスト情報 141 の復帰動作

コンテキスト復帰部 125 は、コンテキスト保存部 131 に格納されている一定量のコンテキスト情報 141 を復帰させ、もとのタスクの実行が再開される。

【0006】

50 【発明が解決しようとする課題】しかしながら、上記の

従来のタスク切替え装置 1 1 1 は、現在実行されているタスクの種類に関わらず、すべての一定量のコンテキスト情報 1 4 1 をコンテキスト保存部 1 3 1 に保存していた。このため、保存および復元のための処理時間が増大し、かつ多大な容量のコンテキスト保存部 1 3 1 を必要とするという問題点を有している。

【0 0 0 7】すなわち、タスクが切り替えられる前に実行されていたのが、アセンブラから変換されたプログラムである場合と C 言語から変換されたプログラムである場合とでは保存すべき情報量が違うにも関わらず、常に一定の情報を保存していた。

【0 0 0 8】具体的には、C 言語を変換するコンパイラは使用するレジスタ数が規定されておりハードウェアが有するレジスタをすべて使用する訳ではない。一方アセンブラでは作成者がレジスタを自由に使用できる。

【0 0 0 9】従って、C 言語から変換された機械語によるタスクを実行中に割り込みが発生した場合はコンパイラで規定されたレジスタ数だけを情報待避すれば足り、アセンブラから変換された機械語によるタスクを実行中に割り込みが発生した場合はハードウェアが備えるすべてのレジスタの情報を待避しなければならない。

【0 0 1 0】本発明はかかる点に鑑み、タスク切替え時に待避するコンテキスト情報量を最適にすることでコンテキストの保存および復元の処理時間とコンテキスト保存部の使用量とを最小限にすることを目的とする。

【0 0 1 1】

【課題を解決するための手段】この課題を解決するため本発明の請求項 1 記載のタスク切替え装置は、退避が必要なコンテキストを識別し、識別情報を所定の領域に退避するタスク状態識別手段と、前記タスク状態識別手段により識別されたコンテキストを退避するコンテキスト退避手段と、前記所定の領域から識別情報を読み出して復帰に必要なコンテキストを識別する復帰状態識別手段と、前記復帰状態識別手段により識別されたコンテキストを復帰するコンテキスト復帰手段とを備えることを特徴とする。

【0 0 1 2】本発明の請求項 2 記載のタスク切替え装置は、請求項 1 記載のタスク切替え装置において、前記タスク状態識別手段は、タスク切り替えがアセンブラ言語のプログラムでなされたか高級言語のプログラムでなされたかの情報を前記識別情報としたものである。

【0 0 1 3】本発明の請求項 3 記載のタスク切替え装置は、請求項 2 記載のタスク切替え装置において、前記コンテキスト退避手段は、前記タスク状態識別手段によりタスク切り替えが高級言語のプログラムでなされたことが識別された場合、前記高級言語の言語処理系において破壊レジスタと定義されたレジスタを含まないコンテキストを退避するとしたものである。

【0 0 1 4】本発明の請求項 4 記載のタスク切替え装置は、請求項 3 記載のタスク切替え装置において、前記コ

ンテキスト退避手段は、引数または戻り値を格納するレジスタに対して、該レジスタが前記破壊レジスタと定義されたレジスタに含まれる場合であっても、該レジスタを退避するとしたものである。

【0 0 1 5】本発明の請求項 5 記載のタスク切替え装置は、請求項 3 記載のタスク切替え装置において、前記コンテキスト退避手段は、前記タスク状態識別手段によりタスク切り替えが高級言語のプログラムでなされたことが識別された場合であっても、該タスク切替えが各タスクから独立なプログラムからなされた場合は、前記破壊レジスタと定義されたレジスタを退避するとしたものである。

【0 0 1 6】本発明の請求項 6 記載のタスク切替え装置は、請求項 5 記載のタスク切替え装置において、各タスクから独立な前記プログラムは、割り込みプログラムであるとしたものである。

【0 0 1 7】本発明の請求項 7 記載のタスク切替え装置は、請求項 1 記載のタスク切替え装置において、前記識別情報は、退避または復帰するコンテキストにいかなる資源を含むかを、ビット毎に該資源を割り付けた情報または含む資源の状態をエンコードした情報であるとしたものである。

【0 0 1 8】本発明の請求項 8 記載のタスク切替え装置は、請求項 1 から 7 のいずれか 1 項に記載のタスク切替え装置において、前記所定の領域と、前記コンテキスト退避手段によりコンテキストが退避される領域とは、予め定められたコンテキスト保存領域において互いに連続するとしたものである。

【0 0 1 9】本発明の請求項 9 記載のタスク切替え装置は、請求項 1 から 7 のいずれか 1 項に記載のタスク切替え装置において、前記所定の領域は、前記コンテキスト退避手段によりコンテキストが退避される際に、該コンテキストに含まれる資源を所定のデータ長に整列させて退避するときに発生する未使用領域の一部または全部であるとしたものである。

【0 0 2 0】

【発明の実施の形態】以下、本発明の実施形態について、図 1 から図 6 を用いて説明する。

【0 0 2 1】図 1 は、本発明の実施形態におけるタスク切替え装置の構成および使用形態図である。

【0 0 2 2】同図において、タスク切替え装置 1 1 は、タスク状態識別部 2 3 と、コンテキスト待避部 2 4 と、タスク切替部 2 5 と、復帰状態識別部 2 6 と、コンテキスト復帰部 2 7 とから構成され、コンテキスト保存部 3 1 との間でデータの読み書きを行う。

【0 0 2 3】この実施の形態では、タスク状態識別部 2 3 によって現在実行中のタスクがアセンブラから変換された機械語か、C 言語等の高級言語から変換された機械語かを識別し、C 言語等から変換された機械語である場合は保存する情報量を最適とするものである。

【0024】すなわち、アセンブラから変換された機械語によるタスクを実行中に割り込みが発生した場合は、ハードウェアが備えるすべてのレジスタの情報を待避する。一方、C言語等から変換された機械語である場合は、コンパイラが規定するレジスタのみの情報を待避するものである。

【0025】以上の構成を持つ本発明のタスク切替え装置11について、タスク部21とタスク独立部22とを実行させる為に必要なコンテキストをコンテキスト保存部31に保存・復元させる動作を説明する。

(1) タスク状態識別部23のタスク状態の退避動作
タスク状態識別部23は、現在実行中のタスクがアセンブラから変換された機械語又はC言語等から変換された機械語によるものかを区別し、その状態の情報をコンテキスト保存部31の特定された未使用領域に識別情報32として保存すると共に、コンテキスト待避部24にその状態を渡す。

【0026】なお、上記の区別は一般に、アセンブラから変換された機械語とC言語から変換された機械語とは別のメモリ空間に格納されるため、いずれのメモリ空間に格納されているかを認識することにより実現する。

(2) コンテキスト待避部24のコンテキスト保存部31への待避動作

コンテキスト待避部24は、タスク状態識別部23から渡されたタスク状態をもとに、タスク状態に応じた最適なコンテキスト情報33をタスク保存部31に待避する。

(3) 復帰状態識別部26のコンテキスト情報33の復帰動作

復帰状態識別部26は、コンテキスト保存部31の特定領域に保存されている識別情報34を読み出し、コンテキスト保存部31に保存されているコンテキスト情報35の種類を識別すると共に、保存されているコンテキスト情報35の種類に関する情報をコンテキスト復帰部27に渡す。

【0027】これにより、保存する情報量を切り替えているため、必要な情報を復帰させることができる。

(4) コンテキスト復帰部27のコンテキスト保存部31からの復帰動作

コンテキスト復帰部27は、復帰状態識別部26から渡された情報をもとに、コンテキスト保存部31からコンテキスト情報35を読み込み、コンテキスト情報33を復帰させて、次のタスクを実行させる。

【0028】以下に本実施形態のタスク切替え装置をフローチャートおよびコンテキスト保存部の状態を図を用いて説明する。

【0029】まず、図2は、本実施形態のタスク切替え装置が対象とするプロセッサのコンテキストとなり得るレジスタを示した図である。このプロセッサは、いずれも24ビットのデータレジスタD0～D3、アドレスレ

ジスタA0～A3、プログラムカウンタPCと、16ビットのプロセッサ状態語PSWを備える。

(1) タスク部21からのタスク切替え呼び出し
タスク部21は主にユーザのアプリケーションタスクで構成され、関数単位のタスク切替えが行われる。

【0030】コンテキストの退避に関しては、タスク状態識別部23とコンテキスト待避部24とが以下のように作用する。

【0031】図3はコンテキスト情報を待避するプログラムフローチャート、図4はアセンブラ言語でタスク切替装置11を呼び出した場合のコンテキスト保存部の構造を示した図、図5は高級言語でタスク切替装置11を呼び出した場合のコンテキスト保存部の構造を示した図である。

【0032】アセンブラ言語におけるタスク切替え呼び出しの場合(ステップ201から始める)、コンテキスト保存部31に退避領域を確保する(ステップ203)。退避領域は、図2に示す全てのコンテキストを格納するための38バイトを必要とする。次にアセンブラ言語を示す識別情報を保存する(ステップ204)。識別情報は、コンテキスト保存部の先頭から3を加算した1バイト分の領域402のLSBに値0として示される。LSB以外は何でもよいが、判断のしやすさの観点からLSB以外は0とする。そしてアセンブラ言語からの呼び出しのみに退避必要なコンテキスト情報を退避する(ステップ205)。ここでは、データレジスタD0とD1、アドレスレジスタA0の内容がそれぞれコンテキスト8情報～コンテキスト10情報として、領域409～411に退避される。最後にアセンブラ言語と高級言語に必要なコンテキスト情報を退避する(ステップ208)。ここでは、データレジスタD2とD3、アドレスレジスタA1～A3、プログラムカウンタPC、プロセッサ状態語PSWの内容がそれぞれコンテキスト1情報～コンテキスト7情報として、領域401と領域403～408に退避される。以上でコンテキスト情報の退避を完了する。なお、コンテキスト2情報～コンテキスト6情報およびコンテキスト8情報～コンテキスト10情報は、コンテキスト1情報と同様にそれぞれの領域の下位3バイトの位置に保存される。

【0033】C言語などの高級言語におけるタスク切替え呼び出しの場合(ステップ202から始める)、コンテキスト保存部31に退避領域を確保する(ステップ206)。退避領域は、図2に示すコンテキストの内から破壊レジスタとして定義されているデータレジスタD0とD1およびアドレスレジスタA0を除くものを格納するための26バイトしか必要としない。破壊レジスタはプログラムの関数間における値の保証が不要なレジスタとしてコンパイラなどの言語処理系で扱われるレジスタである。次に高級言語を示す識別情報を保存する(ステップ207)。識別情報は、コンテキスト保存部の先頭

から3を加算した1バイト分の領域402のLSBに値1として示される。LSB以外は何でもよいが、判断のしやすさの観点からLSB以外は0とする。そしてアセンブラ言語と高級言語に共通して必要なコンテキスト情報を退避する(ステップ208)。ここでは、データレジスタD2とD3、アドレスレジスタA1~A3、プログラムカウンタPC、プロセッサ状態語PSWの内容がそれぞれコンテキスト1情報~コンテキスト7情報として、領域401と領域403~408に退避される。以上でコンテキスト情報の退避を完了する。なお、コンテキスト2情報~コンテキスト6情報は、コンテキスト1情報と同様にそれぞれの領域の下位3バイトの位置に保存される。

【0034】コンテキストの復帰に関しては、復帰状態識別部26とコンテキスト復帰部27とが以下のように作用する。

【0035】図6は、コンテキスト情報を復帰するプログラムフローチャートである。まず、復帰状態識別部26がコンテキスト保存部の先頭に3を加算した領域402から識別情報を読み出し(ステップ301)、アセンブラ言語からの呼び出しか、高級言語からの呼び出しかを識別する(ステップ302)。領域402の1バイトのデータが0なら前者、1なら後者である。

【0036】アセンブラ言語からの呼び出しと判定した場合は、アセンブラ言語のみ待避したコンテキスト情報を復帰し(ステップ303)、その領域を開放する(ステップ304)。具体的には、図4において、領域409~411からコンテキスト8情報~コンテキスト10情報を取り出してそれぞれデータレジスタD0、アドレスレジスタA0とA1に格納する。続いて共通で待避されたコンテキスト情報を復帰してその領域を開放する(ステップ305)。具体的には、領域401と領域403~408からコンテキスト1情報~コンテキスト7情報を取り出してそれぞれデータレジスタD1~D3、アドレスレジスタA2とA3、プログラムカウンタPC、プロセッサ状態語PSWに格納する。以上でコンテキスト情報の退避を完了する。なお、領域の開放はスタックポインタを更新するなどして実現される。

【0037】高級言語からの呼び出しと判定した場合は、共通で待避されたコンテキスト情報を復帰してその領域を開放する(ステップ305)。破壊レジスタとして定義されているデータレジスタD0およびアドレスレジスタA0とA1については復帰する必要がない。具体的には、領域401と領域403~408からコンテキスト1情報~コンテキスト7情報を取り出してそれぞれデータレジスタD1~D3、アドレスレジスタA2とA3、プログラムカウンタPC、プロセッサ状態語PSWに格納する。以上でコンテキスト情報の退避を完了する。なお、領域の開放はスタックポインタを更新するなどして実現される。

(2) タスク独立部22からのタスク切替え呼び出し
タスク独立部22は主に割り込みプログラムである。従って、高級言語におけるタスク切替え呼び出しであっても全てのコンテキストを退避および復帰する必要がある。アセンブラ言語および高級言語いずれの場合も、コンテキスト保存部の構造は図4に示すものと同じになり、退避および復帰の動作は上記(1)のアセンブラ言語の場合と同一である。

【0038】以上のように本発明の実施形態によれば、タスク切替えがアセンブラ言語からなされたか高級言語からなされたかを示す情報をコンテキスト保存部31の領域402に保存し、高級言語の場合の破壊レジスタの退避を省略する事で、コンテキスト保存部31に保存されるコンテキスト情報量は最適になり、コンテキストの保存および復元の処理時間とコンテキスト保存部31の使用量とを削減することができる。さらに、領域402には、3バイト長のレジスタをワード整列させたときに発生する未使用領域を割り当てているため、識別情報の保存によるコンテキスト保存部31の使用量の増大を招かない。

【0039】なお、上記実施形態では、タスク切替えがアセンブラ言語からなされたか高級言語からなされたかを示す2値の情報のみをコンテキスト保存部31の領域402に保存しているが、直接的にコンテキスト保存部31に保存されるコンテキスト情報を識別するようにしてもよい。例えば、8ビットの領域402のビットそれぞれにデータレジスタD0~D3およびアドレスレジスタA0~A3の退避の有無を示すようにしてもよいし、256種類までの情報を8ビットにエンコードするようにしてもよい。

【0040】また、上記実施形態では、識別情報を格納する領域402には3バイト長のレジスタをワード整列させたときに発生する未使用領域を割り当てているが、プロセッサ状態語PSWを退避する領域の、本来プロセッサ状態語PSWに未使用のビットを利用してもよいし、敢えて専用の領域を割り当ててもよい。後者の場合でも、該専用領域がコンテキスト保存部に保存されるコンテキスト情報の削減量を下回れば効果がある。特に、4バイト長のレジスタの内容を保存する場合には有効となることが多い。

【0041】また、上記実施形態では、破壊レジスタをデータレジスタD0とD1およびアドレスレジスタA0としているが、これらに限定されるものではない。さらに、破壊レジスタに引数または戻り値を格納するレジスタを割り付ける場合、高級言語におけるタスク切替え呼び出しであってもこれらのレジスタを退避や復帰するようにしてもよい。

【0042】

【発明の効果】以上のように本発明の請求項1記載のタスク切替え装置は、退避が必要なコンテキストを識別

し、識別情報を所定の領域に退避するタスク状態識別手段と、前記タスク状態識別手段により識別されたコンテキストを退避するコンテキスト退避手段と、前記所定の領域から識別情報を読み出して復帰が必要なコンテキストを識別する復帰状態識別手段と、前記復帰状態識別手段により識別されたコンテキストを復帰するコンテキスト復帰手段とを備えることを特徴とする。

【0043】この構成によれば、退避および復帰が必要なコンテキストのみが識別され、保存されるコンテキスト情報量は最適になり、コンテキストの量を削減することができる。

【0044】本発明の請求項2記載のタスク切替え装置は、請求項1記載のタスク切替え装置において、前記タスク状態識別手段は、タスク切り替えがアセンブラ言語のプログラムでなされたか高級言語のプログラムでなされたかの情報を前記識別情報としたものである。

【0045】この構成によれば、最小の識別情報のみで退避および復帰が必要なコンテキストを効率よく識別することができる。

【0046】本発明の請求項3記載のタスク切替え装置は、請求項2記載のタスク切替え装置において、前記コンテキスト退避手段は、前記タスク状態識別手段によりタスク切り替えが高級言語のプログラムでなされたことが識別された場合、前記高級言語の言語処理系において破壊レジスタと定義されたレジスタを含まないコンテキストを退避するとしたものである。

【0047】この構成によれば、破壊レジスタの退避および復帰がなくなり、保存されるコンテキスト情報量は最適になり、コンテキストの量を削減することができる。

【0048】本発明の請求項4記載のタスク切替え装置は、請求項3記載のタスク切替え装置において、前記コンテキスト退避手段は、引数または戻り値を格納するレジスタに対して、該レジスタが前記破壊レジスタと定義されたレジスタに含まれる場合であっても、該レジスタを退避するとしたものである。

【0049】この構成によれば、引数または戻り値が格納される破壊レジスタは退避され、容易に矛盾なくコンテキストが切替えられる。

【0050】本発明の請求項5記載のタスク切替え装置は、請求項3記載のタスク切替え装置において、前記コンテキスト退避手段は、前記タスク状態識別手段によりタスク切り替えが高級言語のプログラムでなされたことが識別された場合であっても、該タスク切替えが各タスクから独立なプログラムからなされた場合は、前記破壊レジスタと定義されたレジスタを退避するとしたものである。

【0051】この構成によれば、各タスクから独立なプログラムでは破壊レジスタは退避され、容易に矛盾なくコンテキストが切替えられる。

【0052】本発明の請求項6記載のタスク切替え装置は、請求項5記載のタスク切替え装置において、各タスクから独立な前記プログラムは、割り込みプログラムであるとしたものである。

【0053】この構成によれば、割り込みプログラムでは破壊レジスタは退避され、容易に矛盾なくコンテキストが切替えられる。

【0054】本発明の請求項7記載のタスク切替え装置は、請求項1記載のタスク切替え装置において、前記識別情報は、退避または復帰するコンテキストにいかなる資源を含むかを、ビット毎に該資源を割り付けた情報または含む資源の状態をエンコードした情報であるとしたものである。

【0055】この構成によれば、識別情報を参照するだけで退避および復帰が真に必要なコンテキストを効率よく識別することができる。

【0056】本発明の請求項8記載のタスク切替え装置は、請求項1から7のいずれか1項に記載のタスク切替え装置において、前記所定の領域と、前記コンテキスト退避手段によりコンテキストが退避される領域とは、予め定められたコンテキスト保存領域において互いに連続するとしたものである。

【0057】この構成によれば、領域のつながりにより識別情報とそれによって識別されるコンテキストとの対応が明確になる。

【0058】本発明の請求項9記載のタスク切替え装置は、請求項1から7のいずれか1項に記載のタスク切替え装置において、前記所定の領域は、前記コンテキスト退避手段によりコンテキストが退避される際に、該コンテキストに含まれる資源を所定のデータ長に整列させて退避するとき発生する未使用領域の一部または全部であるとしたものである。

【0059】この構成によれば、識別情報の退避によるコンテキストの量の増大を招かない。

【図面の簡単な説明】

【図1】本発明の実施形態におけるタスク切替え装置の構成および使用形態図

【図2】本実施形態のタスク切替え装置が対象とするプロセッサのコンテキストとなり得るレジスタを示した図

【図3】同実施形態におけるタスク切替え装置のコンテキスト情報を退避するプログラムフローチャート

【図4】アセンブラ言語でタスク切替え装置11を呼び出した場合のコンテキスト保存部の構造を示した図

【図5】高級言語でタスク切替え装置11を呼び出した場合のコンテキスト保存部の構造を示した図

【図6】同実施形態におけるタスク切替え装置のコンテキスト情報を復帰するプログラムフローチャート

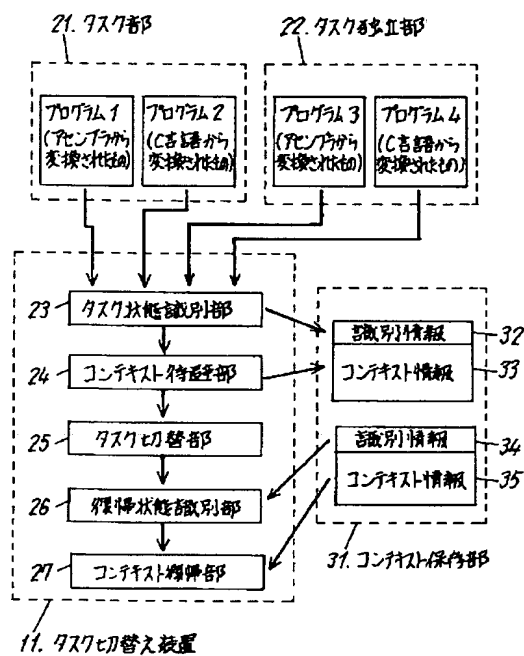
【図7】従来のタスク切替え装置の構成および使用形態図

50 【符号の説明】

11

- 1 1 タスク切替え装置
- 2 1 タスク部
- 2 2 タスク独立部
- 2 3 タスク状態識別部
- 2 4 コンテキスト待避部
- 2 5 タスク切替部
- 2 6 復帰状態識別部
- 2 7 コンテキスト復帰部
- 3 1 コンテキスト保存部
- 3 2、3 4 識別情報

【図 1】

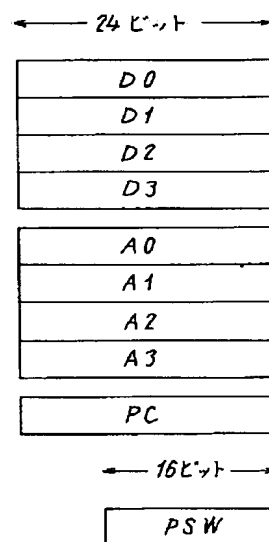


12

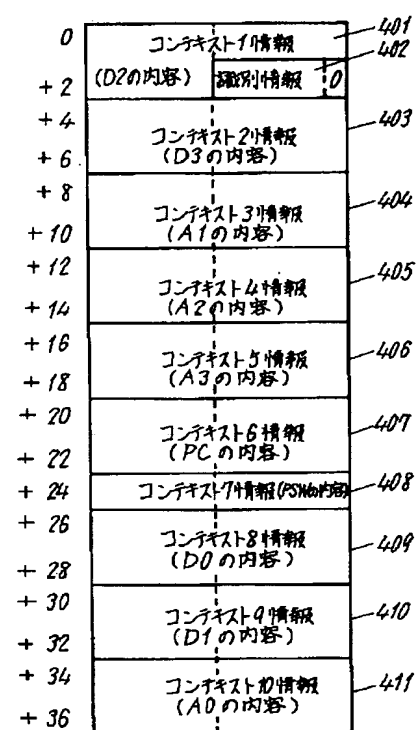
- 3 3、3 5 コンテキスト情報
- 1 1 1 タスク切替え装置
- 1 2 1 タスク部
- 1 2 2 タスク独立部
- 1 2 3 コンテキスト待避部
- 1 2 4 タスク切替部
- 1 2 5 コンテキスト復帰部
- 1 3 1 コンテキスト保存部
- 1 4 1 コンテキスト情報

10

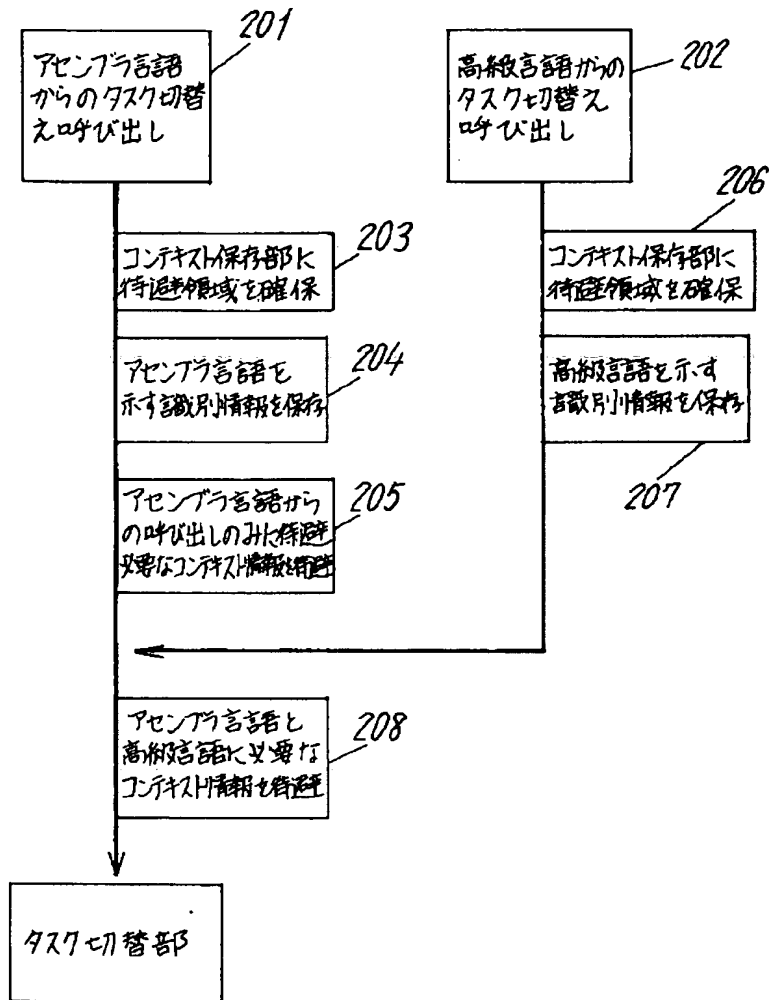
【図 2】



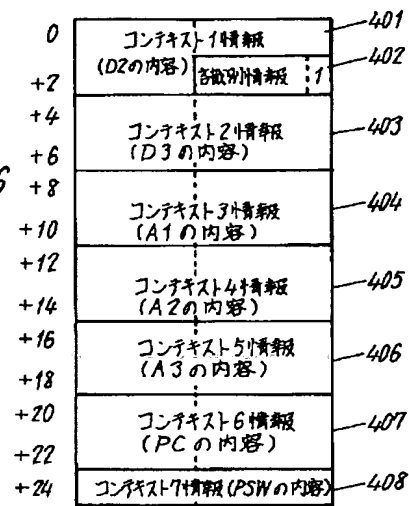
【図 4】



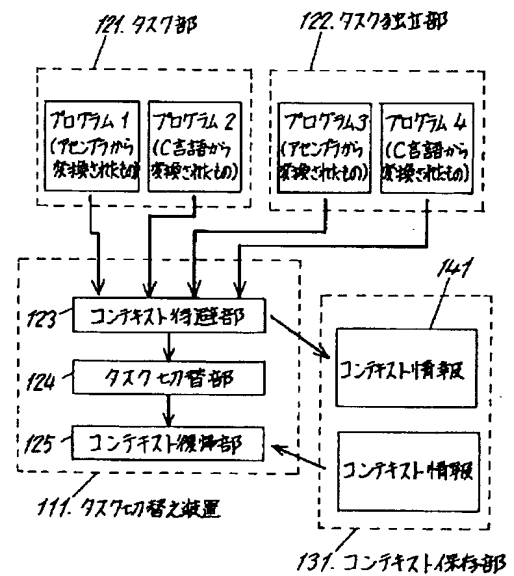
【図 3】



【図 5】



【図 7】



【図6】

